

# Low-Rank Structure Learning via Log-Sum Heuristic Recovery

Yue Deng, Qionghai Dai, *Senior Member, IEEE*, Risheng Liu,  
Zengke Zhang, Sanqing Hu, *Senior Member, IEEE*

**Abstract**—Recovering intrinsic data structure from corrupted observations plays an important role in various tasks in the communities of machine learning and signal processing. In this paper, we propose a novel model, named log-sum heuristic recovery (LHR), to learn the essential low-rank structure from corrupted data. Different from traditional approaches, which directly utilize  $\ell_1$  norm to measure the sparseness, LHR introduces a more reasonable log-sum measurement to enhance the sparsity in both the intrinsic low-rank structure and in the sparse corruptions. Although the proposed LHR optimization is no longer convex, it still can be effectively solved by a majorization-minimization (MM) type algorithm, with which the non-convex objective function is iteratively replaced by its convex surrogate and LHR finally falls into the general framework of reweighted approaches. We prove that the MM-type algorithm can converge to a stationary point after successive iteration. We test the performance of our proposed model by applying it to solve two typical problems: robust principal component analysis (RPCA) and low-rank representation (LRR). For RPCA, we compare LHR with the benchmark Principal Component Pursuit (PCP) method from both the perspectives of simulations and practical applications. For LRR, we apply LHR to compute the low-rank representation matrix for motion segmentation and stock clustering. Experimental results on low rank structure learning demonstrate that the proposed Log-sum based model performs much better than the  $\ell_1$ -based method on for data with higher rank and with denser corruptions.

**Index Terms**—Log-sum heuristic, compressive sensing, sparse optimization, matrix learning, nuclear norm minimization.

## I. INTRODUCTION

Learning the intrinsic data structures via matrix analysis [1][2] has received wide attention in many fields, e.g., neural network[3], learning system[4][5], control theory [6], computer vision [7][8] and pattern recognition [9][10]. There are quite a number of efficient mathematical tools for rank analysis, e.g., Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). However, these typical approaches could only handle some preliminary and simple problems. With the recent progresses of compressive sensing [11], a new concept on nuclear norm optimization has emerged into the field of rank minimization [12] and has led to a number of interesting applications, e.g. low rank structure learning (LRSL) from corruptions.

LRSL is a general model for many practical problems in the communities of machine learning and signal processing, which considers learning a data of the low rank structure from sparse errors [13] [14][15][16]. Such problem can be formulated as:  $\mathbf{P} = f(\mathbf{A}) + g(\mathbf{E})$ , where  $\mathbf{P}$  is the corrupted matrix observed in practical world;  $\mathbf{A}$  and  $\mathbf{E}$  are *low-rank* matrix and *sparse* corruption, respectively and the functions  $f(\cdot)$  and  $g(\cdot)$  are both linear mappings. Recovering two variables (i.e.,  $\mathbf{A}$  and  $\mathbf{E}$ ) just from one equation is an ill-posed problem but is still possible to be addressed by optimizing:

$$(P0) \quad \min_{(\mathbf{A}, \mathbf{E})} \quad rank(\mathbf{A}) + \lambda \|\mathbf{E}\|_{\ell_0} \quad (1)$$

$$s.t. \quad \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}).$$

In (P0),  $rank(\mathbf{A})$  is adopted to describe the low-rank structure of matrix  $\mathbf{A}$ , and the sparse errors are penalized via  $\|\mathbf{E}\|_{\ell_0}$ , where  $\ell_0$  norm counts the number of all the non-zero entries in a matrix. (P0) is always referred as sparse optimization since rank term and  $\ell_0$  norm are sparse measurements for matrices and vectors, respectively. However, such sparse optimization is of little use due to the discrete nature of (P0) and the exact solution to it requires an intractable combinatorial search.

A common approach that makes (P0) trackable tries to minimize its convex envelope, where the rank of a matrix is replaced by the nuclear norm and the sparse errors are penalized via  $\ell_1$  norm, which are convex envelopes for  $rank(\cdot)$  and  $\|\cdot\|_{\ell_0}$ , respectively. Although this framework is developed on two different norms, essentially, it is based on  $\ell_1$  heuristic since nuclear norm can be regarded as a specific case of  $\ell_1$  norm [17]. In practical applications, LRSL via  $\ell_1$  heuristic is powerful enough for many learning tasks with relative low rank structure and sparse corruptions. However, when the desired matrix becomes complicated, e.g., it has high intrinsic rank structure or the corrupted errors become dense, the convex  $\ell_1$  heuristic approaches may not achieve promising performances. In order to handle those tough tasks via LRSL, in this paper we take the advantages of non-convex approximation, rather than the convex  $\ell_1$  heuristic, to better enhance the sparseness of signals.

We propose log-sum heuristic recovery (LHR) to use log-sum term as the basic sparse heuristic functionality for sparse optimization in (P0). There are mainly two reasons that we use such a non-convex term to conduct LRSL. The main reason is due to its sparseness. It is indicated in [18] that log-sum term is a closer approximation to the  $\ell_0$  norm than the  $\ell_1$  norm for sparse vector representation. Therefore, it naturally inspires us to generalize its advantages from vector

Y. Deng, Q. Dai and Z. Zhang are with the Department of Automation, Tsinghua University, Beijing, China (email: daiqh@tsinghua.edu.cn).

R. Liu is with School of Mathematical Sciences, Dalian University of Technology, Dalian, China. (email: rsliu0705@gmail.com).

S. Hu is with College of Computer Science, Hangzhou Dianzi University, Hangzhou, China (email: sqhu@hdu.edu.cn).

recovery to matrix learning. Moreover, although the objective function derived from the log-sum term is non-convex, it is also possible to solve it by convex optimizations because the convex surrogates of log-sum term can be well defined by Taylor expansion. We will introduce an effective non-convex optimization strategy called *majorization-minimization*(MM) [19][20] to solve it next.

MM algorithm is implemented in an iterative way that it first replaces the non-convex component of the objective with a convex upper-bound and then to minimize the convex upper-bound, which exactly makes the non-convex problem fall into the general paradigm of the reweighted schemes. Accordingly, it is possible to solve the non-convex optimization following a sequence of convex optimizations and we will prove that with the MM framework, LHR finally converges to a stationary point after successive iterations.

LHR is a general paradigm for LRSL and we will adapt it to two specific models for practical applications. In a nutshell, LHR will be used to solve the problems of low rank matrix recovery (LRMR) and low rank representation (LRR). In LRMR, LHR is used to recover a low rank matrix from sparse corruptions and its performance will be compared with the benchmark Principle Component Pursuit (PCP) [13] method. In practice, our approach often performs very well in spite of its simplicity. By numerical simulations, LHR could handle many tough tasks that typical algorithm fails to handle. Moreover, the feasible region of LHR is much larger than PCP, which implies that it could deal with much denser corruptions and exhibits much higher rank tolerance. The feasible region of PCP subjects to the boundary of  $\eta^{PCP} + \xi^{PCP} = 0.35$ , where  $\eta$  and  $\xi$  are rank rate and error rate, respectively. With the proposed LHR model, the feasible boundary can be extended to  $\eta^{LHR} + \xi^{LHR} = 0.58$ . The advancements are also verified on two practical applications of shadow removal on face images and video background modeling.

In the second task of low rank representation, the power of LHR model will be generalized to low rank representation for subspace clustering (SC), the goal of which aims at recovering the underlying low rank correlation of subspaces in spite of noisy disturbances. In order to judge the performances, we will first apply it to motion segmentation in video sequences, which is a benchmark test for SC algorithms. Besides, in order to highlight the robustness of LHR to noises and disturbances, we apply LHR to stock clustering that is to determine a stock's industrial category given its historical price record. From both the experiments, LHR gains higher clustering accuracy than other state-of-the-art algorithms and the improvements are especially noticeable on the stock data which includes significant disturbances.

The contributions of this work are three-folds:

- This work presents a log-sum heuristic recovery (LHR) algorithm to handle the typical LRSL problem with an enhanced sparsity term. We introduce a majorization-minimization algorithm to solve the non-convex LHR optimization with reweighted schemes and theoretical justifications are provided to prove that the proposed algorithm converges to a stationary point.
- The proposed LHR model extends the feasible region of

existing  $\ell_1$  norm based LSRL algorithm, which implies that it could successfully handle more learning tasks with denser corruptions and higher rank.

- We apply the LHR model to a new task of stock clustering which serves to demonstrate that low rank structure learning is not only a powerful tool restricted in the areas of image and vision analysis, but also can be applied to solve the profitable financial problems.

The remainder of this paper is organized as follows. We review previous works in Section II. Section III introduces the general LHR model and discusses how to solve the non-convex LHR by MM algorithm. We addresses the low rank matrix recovery (LRMR) problem and compare LHR model with PCP from both the simulations and practical applications in Section IV. The LHR model for low rank representation and subspace segmentation is discussed in Section V. Section VI concludes this paper.

## II. PREVIOUS WORKS

In this part, we review some related works from the following perspectives. First, we discuss two famous models in LRSL, i.e., Low Rank Matrix Recovery (LRMR) from corruptions and low rank representation (LRR). Then, some previous works about Majorization-Minimization algorithm and reweighted approaches are presented.

### A. Low rank structure learning

1) *Low rank matrix recovery*: Corrupted matrix recovery considers decomposing a low rank matrix from sparse corruptions which can be formulated as  $\mathbf{P} = \mathbf{A} + \mathbf{E}$ , where  $\mathbf{A}$  is a low rank matrix,  $\mathbf{E}$  is the sparse error and  $\mathbf{P}$  is the observed data from real world devices, e.g. cameras, sensors and other equipments. The rank of  $\mathbf{P}$  is not low, in most scenarios, due to the disturbances of  $\mathbf{E}$ . How can we recover the low rank structure of the matrix from gross errors? This interesting topic has been discussed in a number of works, e.g. [13] [14] and [15]. Wright *et al.* proposed the PCP (a.k.a. RPCA) to minimize the nuclear norm of a matrix by penalizing the  $\ell_1$  norm of errors [14]. PCP could exactly recover the low rank matrix from sparse corruptions. In some recent works, Ganesh *et al.* investigated the parameter choosing strategy for PCP from both the theoretical justifications and simulations [21]. In this work, we will introduce the reweighted schemes to further improve the performances of PCP. Our algorithm could exactly recover a corrupted matrix from much denser errors and higher rank.

2) *Low rank representation*: Low rank representation[5] is a robust tool for subspace clustering [22], the desired task of which is to classify the mixed data in their corresponding subspaces/clusters. The general model of LRR can be formulated as  $\mathbf{P} = \mathbf{PA} + \mathbf{E}$ , where  $\mathbf{P}$  is the original mixed data,  $\mathbf{A}$  is the affine matrix that reveals the correlations between different pairs of data and  $\mathbf{E}$  is the residual of such a representation. In LRR, the affine matrix  $\mathbf{A}$  is assumed to be low rank and  $\mathbf{E}$  is regarded as sparse corruptions. Compared with existing SC algorithms, LRR is much robust to noises and archives promising clustering results on public datasets. In this work,

inspired by LRR, we will introduce the log-sum recovery paradigm to LRR and show that, with the log-sum heuristic, its robustness to corruptions can be further improved.

### B. MM algorithm and reweighted approaches

Majorization-Minimization (MM) algorithm is widely used in machine learning and signal processing. It is an effective strategy for non-convex problems in which the hard problem is solved by optimizing a series of easy surrogates. Therefore, most optimizations via MM algorithm fall into the framework of reweighted approaches.

In the field of machine learning, MM algorithm has been applied to parameters selection for bayesian classification [23]. In the area of signal processing, MM algorithm leads to a number of interesting applications, including wavelet-based processing [24] and total variation (TV) minimization [25]. For compressive sensing, reweighted method was used in  $\ell_1$  heuristic and led to a number of practical applications including portfolio management [26] and image processing [18]. Reweighted nuclear norm was first discussed in [27] and the convergence of such approach has been proven in [28].

Although there are some previous works on reweighted approaches for rank-minimization, our approach is quite different. First, this work tries to consider a new problem of low rank structure learning from corruptions while not on the single task of sparse signal or nuclear norm minimization. Besides, existing works on reweighted nuclear norm minimization in [27] [28] are solved by semi-definite programming which could only handle the matrix of relative small size. In this paper, we will use the first order numerical algorithm (e.g., alternating direction method (ADM)) to solve the reweighed problem, which can significantly improve the numerical performance. Due to the distributed optimization strategy, it is possible generalize the learning capabilities to large scale matrices.

## III. CORRUPTED MATRIX RECOVERY VIA LOG-SUM HEURISTIC

In this part, we first discuss the widely used  $\ell_1$ -based method for LRS. Then, the log-sum heuristic (LHR) approach is proposed and we introduce the MM algorithm to solve it. Finally, theoretical justifications are presented to prove that LHR can converge to a stationary point by reweighted approaches.

### A. $\ell_1$ heuristic for corrupted low rank matrix recovery

As stated previously, the basic optimization (P0) is non-convex and generally impossible to be solved as its solution usually requires an intractable combinatorial search. In order to make (1) it trackable, convex alternatives are widely used in a number of works, e.g. [13] [14]. Among these approaches, one prevalent method tries to replace the rank of a matrix by its convex envelope, i.e., the nuclear norm and the  $\ell_0$  sparsity is penalized via  $\ell_1$  norm. Accordingly, by convex relaxation,

the problem in (2) can actually be recast as a semi-definite programming.

$$\begin{aligned} \min_{(\mathbf{A}, \mathbf{E})} \quad & \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_{\ell_1} \\ \text{s.t.} \quad & \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}), \end{aligned} \quad (2)$$

where  $\|\mathbf{A}\|_* = \sum_{i=1}^r \sigma_i(A)$ , is the nuclear norm of the matrix which is defined as the summation of the singular values of  $\mathbf{A}$ ; and  $\|\mathbf{E}\|_{\ell_1} = \sum_{ij} |E_{ij}|$  is the  $\ell_1$  norm of a matrix. Although the objective in (2) involves two norms: nuclear norm and  $\ell_1$  norm, its essence is based on the  $\ell_1$  heuristic. We will verify this point with the following lemma.

**Lemma III.1.** *For a matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$ , its nuclear norm is equivalent to the following optimization:*

$$\|\mathbf{X}\|_* = \left\{ \begin{array}{l} \min_{(\mathbf{Y}, \mathbf{Z}, \mathbf{X})} \frac{1}{2} [\text{tr}(\mathbf{Y}) + \text{tr}(\mathbf{Z})] \\ \text{s.t.} \quad \begin{bmatrix} \mathbf{Y} & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Z} \end{bmatrix} \succeq 0, \end{array} \right\} \quad (3)$$

where  $\mathbf{Y} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{Z} \in \mathbb{R}^{n \times n}$  are both symmetric and positive definite. The operator  $\text{tr}(\cdot)$  means the trace of a matrix and  $\succeq$  represents semi-positive definite.

The proof of Lemma.III.1 may refer to [26][12]. According to this lemma, we can replace the nuclear norm in (2) and formulate it in the form of:

$$\begin{aligned} \min_{(\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E})} \quad & \frac{1}{2} [\text{tr}(\mathbf{Y}) + \text{tr}(\mathbf{Z})] + \lambda \|\mathbf{E}\|_{\ell_1} \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{Z} \end{bmatrix} \succeq 0 \\ & \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}). \end{aligned} \quad (4)$$

From Lemma.III.1, we know that both  $\mathbf{Y}$  and  $\mathbf{Z}$  are symmetric and positive definite. Therefore, the trace of  $\mathbf{Y}$  and  $\mathbf{Z}$  can be expressed as a specific form of  $\ell_1$  norm, i.e.  $\text{tr}(\mathbf{Y}) = \|\text{diag}(\mathbf{Y})\|_{\ell_1}$ .  $\text{diag}(\mathbf{Y})$  is an operator that only keeps the entries on the diagonal position of  $\mathbf{Y}$  in a vector. Therefore, the optimization in (4) can be expressed as:

$$\min_{\hat{\mathbf{X}} \in \hat{\mathcal{D}}} \frac{1}{2} (\|\text{diag}(\mathbf{Y})\|_{\ell_1} + \|\text{diag}(\mathbf{Z})\|_{\ell_1}) + \lambda \|\mathbf{E}\|_{\ell_1}, \quad (5)$$

where  $\hat{\mathbf{X}} = \{\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}\}$  and

$$\hat{\mathcal{D}} = \{(\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}) : \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{Z} \end{bmatrix} \succeq 0, (\mathbf{A}, \mathbf{E}) \in C\}.$$

$(\mathbf{A}, \mathbf{E}) \in C$  stands for convex constraint.

### B. Log-sum heuristic for matrix recovery

By Lemma.III.1, the convex problem with two norms in (2) has been successfully converted to an optimization only with  $\ell_1$  norm and therefore it is called  $\ell_1$ -heuristic.  $\ell_1$  norm is the convex envelope of the concave  $\ell_0$  norm but a number of previous research works have indicated the limitation of approximating  $\ell_0$  sparsity with  $\ell_1$  norm, e.g., [18][29]. It is natural to ask, for example, whether might a different alternative not only find a correct solution, but also outperform the performance of  $\ell_1$  norm? Next we will introduce the log-sum term to represent the sparsity of signals.

**Definition III.2.** For any matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$ , the log-sum term is defined as  $\|\mathbf{X}\|_L = \sum_{ij} \log(|X_{ij}| + \delta)$ , where  $\delta > 0$  is a small regularization constant.

The prominent reason that we use this term is mainly due to its sparsity. As indicated in [18], the log-sum term lies between the scope of the  $\ell_0$  norm and  $\ell_1$  norm, which makes it be a closer approximation of  $\ell_0$  norm [18][26] and therefore, it is used to encourage the sparsity in the optimization. We propose Log-sum Heuristic Recovery (LHR) model  $H(\hat{X})$ :

$$(\text{LHR})H(\hat{X}) = \min_{\hat{X} \in \hat{D}} \frac{1}{2} (\|\text{diag}(\mathbf{Y})\|_L + \|\text{diag}(\mathbf{Z})\|_L) + \lambda \|\mathbf{E}\|_L \quad (6)$$

From the formulation of LHR, obviously, it differs from (5) only on the selection of the sparse norm, where the later uses log-sum term instead of the typical  $\ell_1$  norm. Although we have placed a powerful term to enhance the sparsity in LHR model, unfortunately, it also causes non-convexity into the objective function. The LHR model is not convex since the log-function over  $\mathbb{R}_{++} = (\delta, \infty)$  is concave. In most cases, non-convex problem can be extremely hard to solve. Fortunately, the convex upper bound of  $\|\cdot\|_L$  can be easily found and defined. Therefore, we will introduce the majorization-minimization algorithm to solve the LHR optimization.

#### C. The majorization-minimization for LHR optimization

The majorization-minimization (MM) algorithm replaces the hard problem by a sequence of easier ones. It proceeds in an Expectation Maximization (EM)-like fashion by repeating two steps of Majorization and Minimization in an iterative way. During the *Majorization* step, it constructs the convex upper bound of the non-convex objective. In the *Minimization* step, it minimizes the upper bound.

To see how the MM works for LHR, Let's recall the objective function in (6) and make some simple algebra operations:

$$\begin{aligned} & \frac{1}{2} (\|\text{diag}(\mathbf{Y})\|_L + \|\text{diag}(\mathbf{Z})\|_L) + \lambda \|\mathbf{E}\|_L \\ &= \frac{1}{2} [\sum_i \log(Y_{ii} + \delta) + \sum_k \log(Z_{kk} + \delta)] \\ & \quad + \lambda \sum_{ij} \log(|E_{ij}| + \delta) \\ &= \frac{1}{2} [\log \det(\mathbf{Y} + \delta \mathbf{I}_m) + \log \det(\mathbf{Z} + \delta \mathbf{I}_n)] \\ & \quad + \lambda \sum_{ij} \log(|E_{ij}| + \delta) \end{aligned} \quad (7)$$

where  $\mathbf{I}_m \in \mathbb{R}^{m \times m}$  is an identity matrix. It is well known that the concave function is bounded by its first-order Taylor expansion. Therefore, we calculate the convex upper bounds of all the terms in (7). For the term  $\log \det(\mathbf{Y} + \delta \mathbf{I}_m)$ ,

$$\log \det(\mathbf{Y} + \delta \mathbf{I}_m) \leq \log \det(\mathbf{\Gamma}_Y + \delta \mathbf{I}_m) + \text{tr}[(\mathbf{\Gamma}_Y + \delta \mathbf{I}_m)^{-1}(\mathbf{Y} - \mathbf{\Gamma}_Y)]. \quad (8)$$

The inequality in (8) holds for any  $\mathbf{\Gamma}_Y \succ 0$ . Similarly, for any  $(\mathbf{\Gamma}_E)_{ij} > 0$ ,

$$\sum_{ij} \log(|E_{ij}| + \delta) \leq \sum_{ij} [\log[(\mathbf{\Gamma}_E)_{ij} + \delta] + \frac{E_{ij} - (\mathbf{\Gamma}_E)_{ij}}{(\mathbf{\Gamma}_E)_{ij} + \delta}] \quad (9)$$

We replace each term in (7) with the convex upper bound and define  $T(\hat{X}|\hat{\Gamma})$  as the surrogate function after convex

relaxation. Therefore, we can instead optimize the following problem

$$\begin{aligned} \min_{\hat{X} \in \hat{D}} T(\hat{X}|\hat{\Gamma}) &= \frac{1}{2} \text{tr}[(\mathbf{\Gamma}_Y + \delta \mathbf{I}_m)^{-1} \mathbf{Y}] + \frac{1}{2} \text{tr}[(\mathbf{\Gamma}_Z + \delta \mathbf{I}_n)^{-1} \mathbf{Z}] \\ & \quad + \lambda \sum_{ij} (\mathbf{\Gamma}_{E_{ij}} + \delta)^{-1} E_{ij} + \text{const}, \end{aligned} \quad (10)$$

In (10), set  $\hat{X} = \{\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}\}$ , which contains all the variables to be optimized and set  $\hat{\Gamma} = \{\mathbf{\Gamma}_Y, \mathbf{\Gamma}_Z, \mathbf{\Gamma}_E\}$  contains all the parameter matrices. At the end of (10), *const* stands for the constants that are irrelative to  $\{\mathbf{Y}, \mathbf{Z}, \mathbf{A}, \mathbf{E}\}$ . In some previous works of MM algorithms [23] [18] [28], they denote the parameter  $\hat{\Gamma}$  in  $t^{\text{th}}$  iteration with the optimal value of  $\hat{X}$  of the last iteration, i.e.  $\hat{\Gamma} = \hat{X}^t$ . According to the discussions above, we provide the MM algorithm for (LHR) minimization in Algorithm 1. Before elaborately discussing

---

#### Algorithm 1: A MM algorithm for LHR minimization

---

**Initialization :**  $t := 0$   
**1 repeat**  
    **Majorization :**  
    2  $\hat{\Gamma}^t := \hat{X}^t$ ;  
    3 Define convex upper bound  $T(\hat{X}|\hat{\Gamma}^t)$ ;  
    **Minimization:**  
    4  $\hat{X}^{t+1} = \arg \min_{\hat{X} \in \hat{D}} T(\hat{X}|\hat{\Gamma}^t)$ ;  
    5  $t := t + 1$ ;  
**6 until** convergence;

---

how to numerically solve the optimization, we will first discuss some theoretical properties of it.

#### D. Theoretical justifications

In this part, for simplicity, we define the objective function in (6) as  $H(\hat{X})$  and the surrogate function in (10) is defined as  $T(\hat{X}|\hat{\Gamma})$ .  $\hat{X}$  is a set containing all the variables and set  $\hat{\Gamma}$  records the parameter matrices. The convergence property of general MM algorithm was separately distributed on some early mathematical journals [30][31] which is a bit obscure and were not generally read by researchers in the community of computer science. Besides, previous works on MM convergence are almost on the variable selection models. In this paper, we specify it to our LHR model and try to explain it in a plain way. Before discussing the convergence property of LHR, we will first provide two lemmas.

**Lemma III.3.** If set  $\hat{\Gamma}^t := \hat{X}^t$ , MM algorithm could monotonically decrease the non-convex objective function  $H(\hat{X})$ , i.e.  $H(\hat{X}^{t+1}) \leq H(\hat{X}^t)$ .

*Proof:* In order to prove the monotonically decrease property, we can instead prove:

$$H(\hat{X}^{t+1}) \leq T(\hat{X}^{t+1}|\hat{\Gamma}^t) \leq T(\hat{X}^t|\hat{\Gamma}^t) = H(\hat{X}^t). \quad (11)$$

We prove (11) by the following three steps:

(i) The first inequality follows from the argument that  $T(\hat{X}|\hat{\Gamma})$  is the upper-bound of  $H(\hat{X})$ .

(ii) The second inequality holds since the MM algorithm computes  $\hat{X}^{t+1} = \arg \min_{\hat{X}} T(\hat{X}|\hat{\Gamma}^t)$ . The function  $T(\cdot)$  is convex, therefore,  $\hat{X}^{t+1}$  is the unique global minimum. This property guarantees that  $T(\hat{X}^{t+1}|\hat{\Gamma}^{t+1}) < T(\cdot|\hat{\Gamma}^t)$  with any  $\hat{X} \neq \hat{X}^{t+1}$  and  $T(\hat{X}^{t+1}|\hat{\Gamma}^{t+1}) = T(\cdot|\hat{\Gamma}^t)$  if and only if  $\hat{X} = \hat{X}^{t+1}$ .

(iii) The last equality can be easily verified by expanding  $T(\hat{X}^t|\hat{\Gamma}^t)$  and making some simple algebra. The transformation is straightforward and omitted here. ■

**Lemma III.4.** *Let  $\hat{X} = \{\hat{X}^0, \hat{X}^1, \dots, \hat{X}^t, \dots\}$  be a sequence generated by MM framework in Algorithm 1, after successive iterations, such a sequence converges to the same limit point.*

*Proof:* We give a proof by contradiction. We assume that sequence  $\hat{X}$  diverges, which means that  $\lim_{t \rightarrow \infty} \|\hat{X}^{t+1} - \hat{X}^t\|_F \neq 0$ . According to the discussions in Appendix B, we know that there exists a convergent subsequence  $\hat{X}^{t_k}$  converging to  $\phi$ , i.e.  $\lim_{k \rightarrow \infty} \hat{X}^{t_k} = \phi$  and meanwhile, we can construct another convergent subsequence  $\hat{X}^{t_{k+1}}$  that  $\lim_{k \rightarrow \infty} \hat{X}^{t_{k+1}} = \varphi$ . We assume

that  $\phi \neq \varphi$ . Since the convex upper-bound  $T(\cdot|\hat{\Gamma})$  is continuous, we get  $\lim_{k \rightarrow \infty} T(\hat{X}^{t_{k+1}}|\hat{\Gamma}^{t_k}) = T(\lim_{k \rightarrow \infty} \hat{X}^{t_{k+1}}|\hat{\Gamma}^{t_k}) <$

$T(\lim_{k \rightarrow \infty} \hat{X}^{t_k}|\hat{\Gamma}^{t_k}) = \lim_{k \rightarrow \infty} T(\hat{X}^{t_k}|\hat{\Gamma}^{t_k})$  The strict less than operator " $<$ " holds because  $\varphi \neq \phi$ . See (ii) in the proof of Lemma III.3 for details. Therefore, it is straightforward to get the following inequalities:  $\lim_{k \rightarrow \infty} H(\hat{X}^{t_{k+1}}) \leq \lim_{k \rightarrow \infty} T(\hat{X}^{t_{k+1}}|\hat{\Gamma}^{t_k}) < \lim_{k \rightarrow \infty} T(\hat{X}^{t_k}|\hat{\Gamma}^{t_k}) = \lim_{k \rightarrow \infty} H(\hat{X}^{t_k})$ . Accordingly,

$$\lim_{k \rightarrow \infty} H(\hat{X}^{t_{k+1}}) < \lim_{k \rightarrow \infty} H(\hat{X}^{t_k}) \quad (12)$$

Besides, it is obvious that the function of  $H(\cdot)$  in (6) is bounded below, i.e.  $H(\hat{X}) > (mn + m + n) \log \delta$ . Moreover, as proved in Lemma III.3,  $H(\hat{X})$  is monotonically decreasing, which guarantee that  $\lim_{t \rightarrow \infty} H(\hat{X}^t)$  exists, i.e.

$$\begin{aligned} \lim_{k \rightarrow \infty} H(\hat{X}^{t_k}) &= \lim_{t \rightarrow \infty} H(\hat{X}^t) = \lim_{t \rightarrow \infty} H(\hat{X}^{t+1}) \\ &= \lim_{k \rightarrow \infty} H(\hat{X}^{t_{k+1}}) \end{aligned} \quad (13)$$

Obviously, (13) contradicts to (12). Therefore,  $\phi = \varphi$  and we get the conclusion that  $\lim_{t \rightarrow \infty} \|\hat{X}^{t+1} - \hat{X}^t\|_F = 0$ . ■

Based on the two lemmas proved previously, we can give the convergence theorem of the proposed LHR model.

**Theorem III.5.** *With the MM framework, LHR model finally converges to a stationary point.*

*Proof:* As stated in Lemma III.4, the sequences generated by MM algorithm converges to a limitation and here we will first prove that the convergence is a *fixed point*. We define the mapping from  $\hat{X}^k$  to  $\hat{X}^{k+1}$  as  $M(\cdot)$ , and it is straightforward to get,  $\lim_{t \rightarrow \infty} \hat{X}^t = \lim_{t \rightarrow \infty} \hat{X}^{t+1} = \lim_{t \rightarrow \infty} M(\hat{X}^t)$ , which implies that  $\lim_{t \rightarrow \infty} \hat{X}^t = \phi$  is a fixed point. In the MM algorithm, when

constructing the upper-bound, we use the first-order Taylor expansion. It is well known that the convex surrogate  $T(\hat{X}|\hat{\Gamma})$  is tangent to  $H(\hat{X})$  at  $\hat{X}$  by the property of Taylor expansion. Accordingly, the gradient vector of  $T(\hat{X}|\hat{\Gamma})$  and  $H(\hat{X})$  are equal when evaluating at  $\hat{X}$ . Besides, we know that at the fixed point,  $\mathbf{0} \in \nabla_{\hat{X}=\phi} T(\hat{X}|\hat{\Gamma})$  and because it is tangent to  $H(\hat{X})$ , we can directly get that  $\mathbf{0} \in \nabla_{\hat{X}=\phi} H(\hat{X})$  which proves that the convergent fixed point  $\phi$  is also a *stationary point* of  $H(\cdot)$ . ■

In this part, we have shown that with the MM algorithm, LHR model could converge to a stationary point. However, it is impossible to claim that the converged point is the global minimum since the objective function of LHR is not convex. Fortunately, with a good starting point, we can always find a desirable solution by iterative approaches. In this paper, the solution of  $\ell_1$  heuristic model was used as a starting point and it could always lead to a satisfactory result.

### E. Solve LHR via reweighted approaches

To numerically solve the LHR optimization, we remove the constants that are irrelative to  $\mathbf{Y}$ ,  $\mathbf{Z}$  and  $\mathbf{E}$  in  $T(\hat{X}|\hat{\Gamma})$  and get the new convex objective

$$\min \frac{1}{2} [tr(\mathbf{W}_Y^2 \mathbf{Y}) + tr(\mathbf{W}_Z^2 \mathbf{Z})] + \lambda \sum_{ij} (W_E)_{ij} E_{ij}$$

where  $\mathbf{W}_{\mathbf{Y}(\mathbf{Z})} = (\mathbf{\Gamma}_{\mathbf{Y}(\mathbf{Z})} + \delta \mathbf{I}_{m(n)})^{-1/2}$  and  $(W_E)_{ij} = (E_{ij} + \delta)^{-1}, \forall ij$ . It is worth noting that  $tr(\mathbf{W}_Y^2 \mathbf{Y}) = tr(\mathbf{W}_Y \mathbf{Y} \mathbf{W}_Y)$ . Besides, since both  $\mathbf{W}_Y$  and  $\mathbf{W}_Z$  are positive definite, the first constraint in (6) is equivalent to

$$\begin{bmatrix} \mathbf{W}_Y & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_Z \end{bmatrix} \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{W}_Y & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_Z \end{bmatrix} \succeq \mathbf{0}$$

Therefore, after convex relaxation, the optimization in (6) now subjects to

$$\begin{aligned} \min \quad & \frac{1}{2} [tr(\mathbf{W}_Y \mathbf{Y} \mathbf{W}_Y) + tr(\mathbf{W}_Z \mathbf{Z} \mathbf{W}_Z)] + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{W}_Y \mathbf{Y} \mathbf{W}_Y & \mathbf{W}_Y \mathbf{A} \mathbf{W}_Z \\ (\mathbf{W}_Y \mathbf{A} \mathbf{W}_Z)^T & \mathbf{W}_Z \mathbf{Z} \mathbf{W}_Z \end{bmatrix} \succeq \mathbf{0} \\ & \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}) \end{aligned} \quad (14)$$

Here, we apply Lemma III.1 to (14) once again and rewrite the optimization in (14) in the form of the summation of the nuclear norm and  $\ell_1$  norm,

$$\begin{aligned} \min_{(\mathbf{A}, \mathbf{E})} \quad & \|\mathbf{W}_Y \mathbf{A} \mathbf{W}_Z\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} \quad & \mathbf{P} = f(\mathbf{A}) + g(\mathbf{E}) \end{aligned} \quad (15)$$

In (15), the operator  $\odot$  in the error term denotes the component-wise product of two variables, i.e., for  $\mathbf{W}_E$  and  $\mathbf{E}$ :  $(\mathbf{W}_E \odot \mathbf{E})_{ij} = (W_E)_{ij} E_{ij}$ . According to [26], we know that  $\mathbf{Y}^* = \mathbf{U} \Sigma \mathbf{U}^T$  and  $\mathbf{Z}^* = \mathbf{V} \Sigma \mathbf{V}^T$ , if we do singular value decomposition for  $\mathbf{A}^* = \mathbf{U} \Sigma \mathbf{V}^T$ . Accordingly, the weight matrix  $\mathbf{W}_Y = (\mathbf{U} \Sigma \mathbf{U}^T + \delta \mathbf{I}_m)^{-1/2}$  and matrix  $\mathbf{W}_Z = (\mathbf{V} \Sigma \mathbf{V}^T + \delta \mathbf{I}_n)^{-1/2}$ .

Here, based on MM algorithm, we have converted the non-convex LHR optimization to be a sequence of convex reweighted problems. We call it *reweighted method* (15) since in each iteration we should re-denote the weight matrix set

$\hat{W}$  and use the updated weights to construct the surrogate convex function. Besides, the objective in (15) is convex with a summation of a nuclear norm and a  $\ell_1$  norm and can be solved by convex optimization. In the next two sections, the general LHR model will be adapted to two specific models and we will provide the optimization strategies for those two models, respectively.

#### IV. LOW RANK MATRIX RECOVERY FROM CORRUPTIONS

In this part, we first apply the LHR model to recover a low rank matrix from corruption and its performance is compared with the widely used Principal Component Pursuit (PCP).

##### A. Joint optimization for LHR

Based on the LHR derivations, the corrupted low rank matrix recovery problem can be formulated as a reweighted problem:

$$\begin{aligned} \min_{(\mathbf{A}, \mathbf{E})} & \|\mathbf{W}_Y \mathbf{A} \mathbf{W}_Z\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} & \mathbf{P} = \mathbf{A} + \mathbf{E} \end{aligned} \quad (16)$$

Due to the reweighted weights are placed in the nuclear norm, it is impossible to directly get the closed-form solution of the nuclear norm minimization. Therefore, inspired by the work [5], we introduce another variable  $\mathbf{J}$  to (16) by adding another equality constraint and to solve,

$$\begin{aligned} \min & \|\mathbf{J}\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} & \mathbf{h}_1 = \mathbf{P} - \mathbf{A} - \mathbf{E} = \mathbf{0} \\ & \mathbf{h}_2 = \mathbf{J} - \mathbf{W}_Y \mathbf{A} \mathbf{W}_Z = \mathbf{0} \end{aligned} \quad (17)$$

Based on the transformation, there is only one single  $\mathbf{J}$  in the nuclear norm of the objective that we can directly get its closed-form update rule by [17]. There are quite a number of methods that can be used to solve it, e.g. with Proximal Gradient (PG) algorithm [32] or Alternating Direction Methods (ADM) [33]. In this paper, we will introduce the ADM method since it is more effective and efficient. Using the ALM method [34], it is computationally expedient to relax the equality in (17) and instead solve:

$$\begin{aligned} L = & \|\mathbf{J}\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} + \langle \mathbf{C}_1, \mathbf{h}_1 \rangle \\ & + \langle \mathbf{C}_2, \mathbf{h}_2 \rangle + \frac{\mu}{2} (\|\mathbf{h}_1\|_F^2 + \|\mathbf{h}_2\|_F^2) \end{aligned} \quad (18)$$

where  $\langle, \rangle$  is an inner product and  $\mathbf{C}_1$  and  $\mathbf{C}_2$  are the lagrange multipliers, which can be updated via dual ascending method. (18) contains three variables, i.e.,  $\mathbf{J}$ ,  $\mathbf{E}$  and  $\mathbf{A}$ . Accordingly, it is possible to solve problem via a distributed optimization strategy called Alternating Direction Method (ADM). The convergence of the ADM for convex problems has been widely discussed in a number of works [33][35]. By ADM, the joint optimization can be minimized by four steps as  $\mathbf{E}$ -minimization,  $\mathbf{J}$ -minimization,  $\mathbf{A}$ -minimization and dual ascending. We first provide the update rule for  $\mathbf{E}$ -minimization,

$$\mathbf{E} = \arg \min_{\mathbf{E}} \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_* + \frac{\mu}{2} \|\mathbf{E} - (\mathbf{P} - \mathbf{A} + \mu^{-1} \mathbf{C}_2)\|_F^2 \quad (19)$$

It is well known (see, for example, [36]) that for scalars  $x$  and  $y$ , the unique optimal solution to the problem

$$\min_x \alpha |x| + \frac{1}{2} (x - y)^2 \quad (20)$$

is given by

$$x^* = \text{sgn}(y) \max(|y| - \alpha, 0) \doteq s_\alpha(y). \quad (21)$$

$\mathbf{E}^*$  is a solution to the  $\mathbf{E}$ -minimization if and only if for all  $i, j$ ,

$$E_{ij}^* = s_{\lambda \mu^{-1} |W_{ij}|} (P - A - \mu^{-1} C)_{ij} \quad (22)$$

Similarly,  $\mathbf{J}$ -minimization can be solved by

$$\mathbf{J} = \arg \min_{\mathbf{J}} \|\mathbf{J}\|_* + \frac{\mu}{2} \|\mathbf{J} - \mathbf{W}_Y \mathbf{A} \mathbf{W}_Z + \mu^{-1} \mathbf{C}_1\|_F^2 \quad (23)$$

For matrices  $\mathbf{X}, \mathbf{D}$ , previous works, e.g. [12] [37], have indicated that the unique closed-form optimal solution to the problem

$$\min_{\mathbf{X}} \alpha \|\mathbf{X}\|_* + \frac{1}{2} \|\mathbf{X} - \mathbf{D}\|_F^2 \quad (24)$$

is given by

$$\mathbf{X}^* = \mathbf{U} s_\alpha(\Sigma) \mathbf{V}^T \doteq d_\alpha(\mathbf{D}), \quad (25)$$

where  $\mathbf{D} = \mathbf{U} \Sigma \mathbf{V}^T$  denotes the singular value decomposition of  $\mathbf{D}$ . From (25), it is immediate that the unique optimal solution to (23) is given by

$$\mathbf{J}^* = d_{\mu^{-1}} (\mathbf{W}_Y \mathbf{A} \mathbf{W}_Z + \mu^{-1} \mathbf{C}_2). \quad (26)$$

Finally, the solution to  $\mathbf{A}$  is based on the following optimization problem,

$$\mathbf{A}^* = \arg \min_{\mathbf{A}} \|\mathbf{h}_1 + \mu^{-1} \mathbf{C}_1\|_F^2 + \|\mathbf{h}_2 + \mu^{-1} \mathbf{C}_2\|_F^2 \quad (27)$$

which is only a summation of two  $F$ -norms that can be addressed by gradient-descending method. Here, we provide the update-rule that

$$\mathbf{A}^{k+1} = \mathbf{A}^k + \gamma [\mathbf{W}_Y (\mathbf{h}_1^k + \mu^{-1} \mathbf{C}_1) \mathbf{W}_Z + (\mathbf{h}_2^k + \mu^{-1} \mathbf{C}_2)]$$

Based on all the previous discussions, we can now give the whole framework to solve the LHR model for LRMR via reweighted schemes in Algorithm 2.

##### B. Numerical simulations

We have explained how to recover a low rank matrix via LHR in preceding sections. In this section, we will conduct some experiments to test its performances with the comparisons to robust PCP from both the simulations and practical data.

TABLE I  
EVALUATIONS OF LOW-RANK MATRIX RECOVERY OF ROBUST PCA AND LOG-SUM HEURISTIC RECOVERY.

$m = n$	methods	$rank(\mathbf{A}^*) = 0.4m \quad \ \mathbf{E}^*\ _{\ell_0} = 0.1m^2$				$rank(\mathbf{A}^*) = 0.1m \quad \ \mathbf{E}^*\ _{\ell_0} = 0.4m^2$			
		$\frac{\ \mathbf{A} - \mathbf{A}^*\ _F}{\ \mathbf{A}^*\ _F}$	$rank(\mathbf{A})$	$\ \mathbf{E}\ _{\ell_0}$	$time(s)$	$\frac{\ \mathbf{A} - \mathbf{A}^*\ _F}{\ \mathbf{A}^*\ _F}$	$rank(\mathbf{A})$	$\ \mathbf{E}\ _{\ell_0}$	$time(s)$
200	PCP	4.6e-1	103	21066	5.9	1.2e-1	107	23098	7.4
	LHR	8.1e-4	80	4000	12.7	1.3e-3	20	16031	14.1
400	PCP	4.3e-1	207	83954	26.1	7.0e-1	214	89370	35.7
	LHR	8.2e-4	160	16218	63.4	1.7e-3	40	47999	54.3
800	PCP	4.8e-1	414	336188	36.2	9.3e-2	348	355878	48.2
	LHR	9.9e-4	20	64283	91.7	2.1e-3	320	191998	108.2

**Algorithm 2:** Optimization strategy of LHR for corrupted matrix recovery

---

**Input** : Corrupted matrix  $P$  and parameter  $\lambda$   
**Initialization** :  $t := 1, E_{ij}^0 := 1, \forall i, j. \mathbf{W}_{Y(Z)}^{(1)} = \mathbf{I}_{m(n)}.$

1 **repeat**  
2   // Dynamically update the weight matrix.  
3    $\mathbf{W}_E^{(t)} := (|\mathbf{E}^{(t-1)}| + \delta_1)^{-1};$   
4    $\mathbf{U}\Sigma\mathbf{V}^T = SVD(\mathbf{A}^{(t-1)});$   
5    $\mathbf{W}_Y^{(t)} := (\mathbf{U}\Sigma\mathbf{U}^T + \delta_2\mathbf{I}_m)^{-1/2};$   
6    $\mathbf{W}_Z^{(t)} := (\mathbf{V}\Sigma\mathbf{V}^T + \delta_2\mathbf{I}_n)^{-1/2};$   
7   Reset  $C_0 > 0; \mu_0 > 0; \rho > 1; k = 1; \mathbf{A}^0 = \mathbf{E}^0 = \mathbf{0};$   
8   **while not converged do**  
9     // Variables updating.  
10     $E_{ij}^k = s_{\lambda\mu^{-1}}(|(W_E^{(t)})_{ij}|)(P - A^{k-1} - \mu^{-1}C_1^k)_{ij}, \forall ij;$   
11     $\mathbf{J}^k = d_{\mu^{-1}}(\mathbf{W}_Y^{(t)}\mathbf{A}^{k-1}\mathbf{W}_Z^{(t)} + \mu^{-1}\mathbf{C}_2^k);$   
12     $\mathbf{A}^k = \mathbf{A}^{k-1} + \gamma[-\mathbf{W}_Y^{(t)}(\mathbf{h}_1^k + \mu^{-1}\mathbf{C}_2^k)\mathbf{W}_Z^{(t)} + (\mathbf{h}_2^k + \mu^{-1}\mathbf{C}_1^k)];$   
13    // Dual ascending.  
14     $\mathbf{C}_1^k = \mathbf{C}_1^{k-1} + \mu_k\mathbf{h}_1^k;$   
15     $\mathbf{C}_2^k = \mathbf{C}_2^{k-1} + \mu_k\mathbf{h}_2^k;$   
16     $k := k + 1, \mu_{k+1} = \rho\mu_k;$   
17   **end**  
18    $(\mathbf{A}^{(t)}, \mathbf{E}^{(t)}) = (\mathbf{A}_k, \mathbf{E}_k);$   
19    $t := t + 1;$   
20 **until convergence;**  
**Output** :  $(\mathbf{A}^{(t)}, \mathbf{E}^{(t)}).$

---

1) *General evaluation:* We demonstrate the accuracy of the proposed LHR algorithm on randomly generated matrices. For an equivalent comparison, we adopted the same data generating method in [13] that all the algorithms are performed on the squared matrices and the ground-truth low rank matrix (rank  $r$ ) with  $m \times n$  entries, denoted as  $\mathbf{A}^*$ , is generated by independent random orthogonal model [13]; the sparse error  $\mathbf{E}^*$  is generated via uniformly sampling the matrix and the error values are randomly generate in the range  $[-100, 100]$ . The corrupted matrix is generated by  $\mathbf{P} = \mathbf{A}^* + \mathbf{E}^*$ , where  $\mathbf{A}^*$  and  $\mathbf{E}^*$  are the ground truth. For simplicity, we denote the rank rate as  $\eta = \frac{rank(\mathbf{A}^*)}{\max\{m, n\}}$  and the error rate as  $\xi = \frac{\|\mathbf{E}\|_{\ell_0}}{m \times n}$ .

For an equivalent comparison, we use the code in [38] to

solve the PCP problem<sup>1</sup>. [13] indicated that PCP method could exactly recover a low rank matrix from corruptions within the region of  $\eta + \xi < 0.35$ . Here, in order to highlight the effectiveness of our LHR model, we directly consider much difficult tasks that we set  $\eta + \xi = 0.5$ . Each experiment is repeated for ten times and the median values<sup>2</sup> are tabulated in Table.I. In the table,  $\frac{\|\mathbf{A} - \mathbf{A}^*\|_F}{\|\mathbf{A}^*\|_F}$  denotes the recovery accuracy,  $rank$  denotes the rank of the recovered matrix  $\mathbf{A}$ ,  $\|\mathbf{E}\|_{\ell_0}$  is the card of the recovered errors and  $time$  records the computational costs (in seconds).

From the results, obviously, compared with PCP, LHR model could exactly recover the matrix from higher ranks and denser errors. However, the table just provides two discrete tests. We will provide more thorough investigation in the next subsection.

2) *Feasible region:* Since the basic optimization involves two terms, i.e., low rank matrix and sparse error. In this part, we will varies these two variables to test the feasible boundary of PCP and LHR, respectively. The experiments are conducted on the  $400 \times 400$  matrices with sparse errors uniformly distributed in  $[-100, 100]$ . In the feasible region verification, when the recovery accuracy is larger than 1% (i.e.,  $\frac{\|\mathbf{A} - \mathbf{A}^*\|_F}{\|\mathbf{A}^*\|_F} > 0.01$ ), it is believed that the algorithm diverges. The two rates  $\eta$  and  $\xi$  are varied from zero to one with the step of 0.025. On each test point, both the PCP and LHR are repeated for 10 times. If the median recovery accuracy is less than 1%, the point is regarded as the feasible point. The feasible regions of these two algorithms are shown in Fig.1(a).

From Fig.1(a), the feasible region of LHR is much larger than the region of PCP. We get the same conclusion as made in [13] that the feasible boundary of PCP roughly fits the curve that  $\eta^{PCP} + \xi^{PCP} = 0.35$ . The boundary of LHR is around the curve that  $\eta^{LHR} + \xi^{LHR} = 0.575$ . Moreover, on the two sides of the red curve in Fig.1(a), the boundary equation can be even extended to  $\eta^{LHR} + \rho^{LHR} = 0.6$ . From this test, it is apparent that the proposed LHR algorithm covers a larger area of the feasible region, which implies that LHR could handle more difficult tasks that robust PCA fails to do.

3) *Convergence discussions:* Finally, we will experimentally verify the convergence of the LHR. The experiments are conducted on  $400 \times 400$  matrices with the rank equivalent to 40 and the portion of gross errors are set as 15%, 30%

<sup>1</sup> In [34], Lin *et al.* provided two solvers, i.e. exact and inexact solvers, to solve the PCP problem. In this paper, we use the exact solver for PCP because it performs better than inexact solver.

<sup>2</sup> We do not use the average values here since in cases of divergence some extreme large outliers may greatly affect the average values of the accuracy.

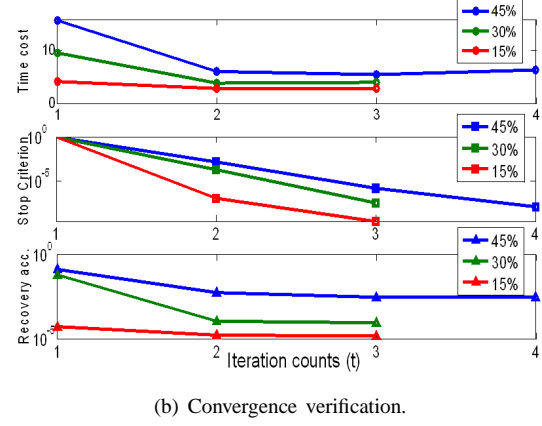
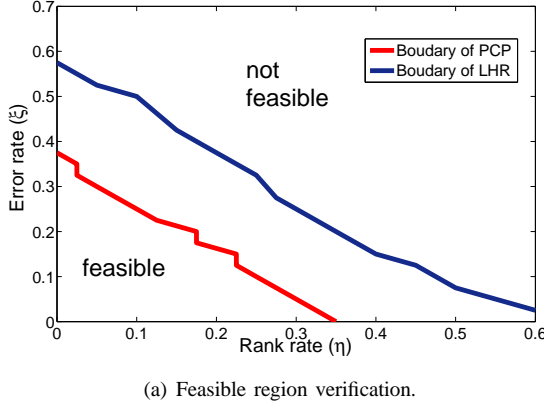


Fig. 1. Feasible region and the convergence verifications.

and 45%, respectively. The experimental results are reported in Fig.IV-B2 where the axis's coordinate denotes the iteration sequences, i.e. the count  $t$  in Algorithm.1.

The top sub-figure in Fig.IV-B2 reports the time cost of each iteration. It is interesting to note that the denser the error is, the more time cost is required for one iteration. Besides, the most time consuming part occurs in the first iteration. During the first iteration, (15) subjects to the typical PCP problem. However, in the second and the third iteration, the weight matrix is assigned with different values and thus it could make (15) converge with less iterations. Therefore, the time cost for each iteration is different in LHR. The first iteration needs many computational resources while the later ones can be further accelerated owing to the penalty of the weight matrix.

The middle sub-figure records the stopping criterion, which is denoted as  $\frac{\|W^{(t+1)} - W^{(t)}\|_F}{\|W^{(t)}\|_F}$ . It is believed that the LHR converges when the stopping criterion is less than  $1e-5$ . It is apparent from Fig.IV-B2 that the LHR could converge in just three iterations with 15% and 30% gross errors. While for the complicated case with 45% errors, LHR can converge in four steps. The bottom sub-figure shows the recovery accuracy after each iteration. It is obvious that the recovery accuracy increases significantly from the first iteration to the second one. Such an increase phenomenon verify the advantage of the reweighted approach derived from LHR.

### C. Practical applications

PCP is a powerful tool for many practical applications. Here, we will conduct two practical applications to verify the effectiveness of PCP and LHR on real-world data.

1) *Shadow and specularities removal from faces*: Following the framework suggested in [13], we stack the faces of the same subject under different lighting conditions as the columns in a matrix  $\mathbf{P}$ . The experiments are conducted on extended Yale-B dataset where each face is with the resolutions of  $192 \times 168$ . Then, the corrupted matrix  $\mathbf{P}$  is recovered by PCP and LHR, respectively. After recovery, the shadows, specularities and other reflectance are removed in the error matrix ( $\mathbf{E}$ ) and the clean faces are accumulated in the low rank matrix ( $\mathbf{A}$ ). The experimental results are provided in Fig.2, where in each sub-figure from left to right are: original

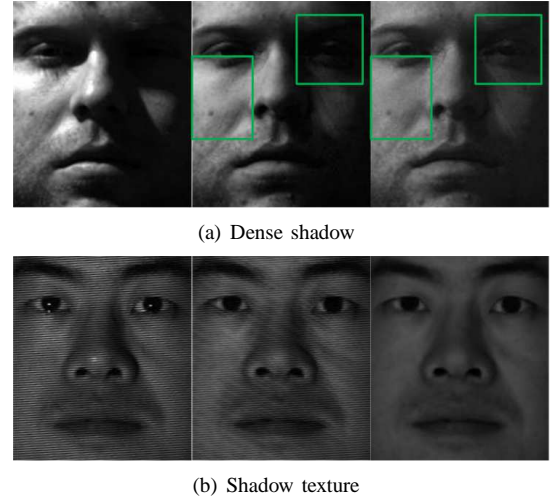


Fig. 2. Shadow and specularities removal from faces (best viewed on screen).

faces in Yale-B (left), faces recovered by PCP (median) and faces recovered by LHR (right), respectively. It is greatly recommended to enlarge the faces in Fig.2 to view the details. In Fig.2(a), when there exist dense shadows on the face image, the effectiveness of LHR becomes apparent to remove the dense shadows distribute on the left face. The dense texture removal ability is especially highlighted in Fig.2(b), where there are significant visual contrasts between the faces recovered by PCP and LHR. The face recovered by LHR is much clean.

2) *Video surveillance*: The background modeling can also be categorized as a low rank matrix recovery problem, where the backgrounds correspond to the low rank matrix  $\mathbf{A}$  and the foregrounds are removed in the error matrix  $\mathbf{E}$ . We use the videos and ground truth in [39] for quantitative evaluations. Three videos used in this experiment are listed in Fig.3.

For the sake of computational efficiency, we normalize each image to the resolutions of  $120 \times 160$  and all the frames are converted to gray-scaler. The benchmark videos used here contain too many frames which lead to a large matrix. It is theoretical feasible to use the two methods for any large matrix recovery. Unfortunately, for practical implementation, large



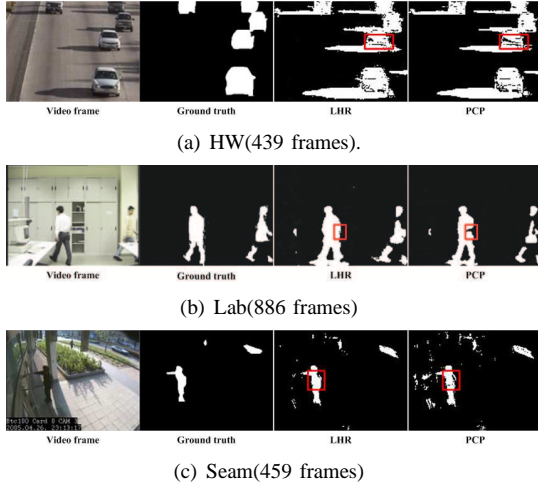


Fig. 3. Benchmark videos for background modeling. In each sub-figure, from left to right are original video frames, foreground ground truth, LHR result and PCP result, respectively.

matrices are always beyond the memory limitation of Matlab. Therefore, for each video, we uniformly divide the large matrix to be sub-matrices which has less than 200 columns. We recover these sub-matrices by setting  $\lambda = \frac{1}{\sqrt{m}}$ , respectively.

The segmented foregrounds and the ground truth are shown in Fig.3. From the results we know that LHR could remove much denser errors from the corrupted matrix rather than PCP. Such claim is verified from three sequences in Fig.3 that LHR makes much complete object recovery from the video. Besides, in Fig.3(c), it is also apparent that LHR only keeps dense errors in the sparse error term. In the seam sequences, there are obvious illumination changes in different frames. PCP is sensitive to these small variations and thus makes much more small isolated noise parts in the foreground. On the other hand, LHR is much robust to these local variations and only keeps dense corruptions in the sparse term.

Although there are many advanced techniques for video background modeling, it is not the main concern of this work. Therefore, without the loss of generality, we use the Mixture of Gaussian (MoG)[40] as the comparison baseline. For evaluation, both the false negative rate (FNR) and false positive rate (FPR) are calculated in the sense of foreground detection. These two scores exactly correspond to the Type I and Type II errors in machine learning, whose definitions may refer to [41]. FNR indicates the ability of the method to correctly recover the foreground and FPR represents the potential of a method on distinguishing the background. Both these two rates are judged by the criterion that the less the better. The experimental results are tabulated in table.IV-C2. We also report the time cost (in minutes) of PCP and LHR on these videos. But we omit the time cost of MoG since it can be finished in almost real time.

From the results, PCP and LHR greatly outperform the performance of MoG. Moreover, LHR has lower FNRs than PCP which implies that LHR could better detect the foreground than PCP. However, on the video highway and seam, the FPR score of LHR is a little worse than PCP. One possible reason may ascribe to that there are too many moving shadows in

TABLE II  
QUANTITATIVE EVALUATION OF PCP AND LHR FOR VIDEO SURVEILLANCE.

Data	False Negative Rate%			False Positive Rate%			Time(m)	
	MoG	PCP	LHR	MoG	PCP	LHR	PCP	LHR
HW	22.2	18.7	14.3	8.8	7.8	8.4	13.2	23.5
Lab.	15.1	10.1	8.3	6.7	6.4	6.1	25.4	43.7
Seam	23.5	11.3	9.2	9.7	6.1	6.3	11.4	19.9

these two videos, where both the objects and shadows are regarded as errors. In the ground truth frames, the shadows are regarded as background. LHR could recover much denser errors from a low rank matrix and thus causes a comparable low FNR score.

## V. LHR FOR LOW RANK REPRESENTATION

In this part, LHR will be applied to the task of low rank representation (LRR)[5][16] by formulating the constraint as  $\mathbf{P} = \mathbf{P}\mathbf{A} + \mathbf{E}$ , where the correlation affine matrix  $\mathbf{A}$  is low rank and the noises in  $\mathbf{E}$  are sparse. In the remaining parts of this section, we will first show how to use the joint optimization strategy to solve the LRR problem by LHR model. Then, two practical applications on motion segmentation and stock clustering will be presented and discussed.

### A. Joint optimization strategy of LHR for LRR

When applying LHR to low rank representation, we should solve a sequence of convex optimizations in the form,

$$\begin{aligned} \min. & \quad \|\mathbf{W}_Y \mathbf{A} \mathbf{W}_Z\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} & \quad \mathbf{P} = \mathbf{P}\mathbf{A} + \mathbf{E} \end{aligned} \quad (28)$$

To make the nuclear norm trackable, we add an equality and tries to solve

$$\begin{aligned} \min. & \quad \|\mathbf{J}\|_* + \lambda \|\mathbf{W}_E \odot \mathbf{E}\|_{\ell_1} \\ \text{s.t.} & \quad \mathbf{b}_1 = \mathbf{P} - \mathbf{P}\mathbf{A} - \mathbf{E} = \mathbf{0} \\ & \quad \mathbf{b}_2 = \mathbf{J} - \mathbf{W}_Y \mathbf{A} \mathbf{W}_Z = \mathbf{0} \end{aligned} \quad (29)$$

Using the ADM strategy and following the similar derivations introduced in subsection IV-A, we can solve the optimization in (29) and we directly provide the update rules for each variable in algorithm 3. To show LHR ideally represents low

---

#### Algorithm 3: Update rule for the variables in (29)

---

- 1  $E_{ij}^k = s_{\lambda\mu^{-1}}[(W_E^{(t)})_{ij}](P - P A^{k-1} - \mu^{-1} C_1^k)_{ij}, \forall i, j;$
  - 2  $\mathbf{J}^k = d_{\mu^{-1}}(\mathbf{W}_Y^{(t)} \mathbf{A}^{k-1} \mathbf{W}_Z^{(t)} + \mu^{-1} \mathbf{C}_2^k);$
  - 3  $\mathbf{A}^k =$   
 $\mathbf{A}^{k-1} + \gamma[\mathbf{W}_Y^{(t)}(\mathbf{b}_1^k + \mu^{-1} \mathbf{C}_2^k) \mathbf{W}_Z^{(t)} + \mathbf{P}^T(\mathbf{b}_2^k + \mu^{-1} \mathbf{C}_1^k)];$
  - 4 // Dual ascending.
  - 5  $\mathbf{C}_1^k = \mathbf{C}_1^{k-1} + \mu_k \mathbf{b}_1^k;$
  - 6  $\mathbf{C}_2^k = \mathbf{C}_2^{k-1} + \mu_k \mathbf{b}_2^k;$
- 

rank structures from data, experiments on subspace clustering are conducted on two datasets. First, we test LHR on slightly corrupted data-set, i.e., Hopkins156 motion database. Since the

effectiveness of LHR are especially emphasized on the data with great corruptions. We will also consider one practical application of using LHR for stock clustering.

### B. Motion segmentation in video sequences

In this part, we apply LHR to the task of motion segmentation in Hopkins155 dataset [22]. Hopkins155 database is a benchmark platform to evaluate general subspace clustering algorithms, which contains 156 video sequences and each of them has been summarized to be a matrix recoding 39 to 50 data vectors. The primary task of subspace clustering is to categorize each motion to its corresponding subspace, where each video corresponds to a sole clustering task and it leads to 156 clustering tasks in total.

For comparisons, we will compare LHR with LRR as well as other benchmark algorithms for subspace clustering. The comparisons include Random Sample Consensus (RANSAC)[42], Generalized Principal Component Analysis (GPCA)[43], Local Subspace Affinity (LSA), Locally Linear Manifold Clustering (LLMC) and Sparse Subspace Clustering (SSC). RANSAC is a statistic method which clusters data by iteratively distinguishing the data by inliers and outliers. GPCA presents an algebraic method to cluster the mixed data by the normal vectors of the data points. Manifold based algorithms, e.g. LSA and LLMC, assume that one point and its neighbors span as a linear subspace and they are clustered via spectral embedding. SSC assumes that the affine matrix between data are sparse and it segments the data via normalized cut[44].

In LRR [5], Liu *et al.* introduced two models that respectively used  $\ell_1$  norm and  $\ell_{2,1}$  norm to penalize sparse corruptions. In this paper, we will only report the results with the comparison to  $\ell_{2,1}$  norm since it always performs better than  $\ell_1$  penalty in LRR. In order to provide a thorough comparison with LRR, we strictly follow the steps and the default parameter settings suggested in [5]. For LHR model, we choose parameter  $\lambda = 0.4$ . In the experiments of LRR for motion segmentation, some post-processing are performed on the learned low rank structure to seek for the best clustering accuracy. For example, in LRR, after getting the representation matrix  $\mathbf{A}$ , an extra PCP processing are implemented on  $\mathbf{A}$  to enhance the low-rankness and such post-processing definitely increases SC accuracy. However, the main contribution of this work only focus LHR model on low-rank structure learning while not on the single task of subspace clustering. Therefore, we exclude all the post-processing steps to emphasize the effectiveness of the LRSL model itself. In our result, all the methods are implemented with the same criterion to avoid bias treatments.

Hopkins155 contains two subspace conditions in a video sequence, i.e., with two motions or three motions and thus we report the segmenting errors for two subspaces (TWO), three subspaces (THREE) and for both conditions (ALL) in Table.V-B. From the results we know that sparse based methods generally outperform other algorithms for motion segmentation. Among three sparse methods, LHR gains the best clustering accuracy. However, the accuracy only has slight

improvements on LRR. As indicated in [5], motion data only contains small corruptions and LRR could already achieve promising performance with the accuracy higher than 90%. With some post-processing implementations, the accuracy can even be further improved. Therefore, in order to highlight the effectiveness of LHR on low rank representation with corrupted data, some more complicated problems will be considered.

TABLE III  
MOTION SEGMENTATION ERRORS (MEAN) OF SEVERAL ALGORITHMS ON THE HOPKINS155 MOTION SEGMENTATION DATABASE.

CATEGORY	METHOD	TWO	THREE	ALL
<i>Algebraic</i>	GPCA	11.2	27.7	14.2
<i>Statistic</i>	RANSAC	8.9	24.1	12.5
<i>Manifold</i>	LSA	8.7	21.4	11.6
	LLMC	8.1	20.8	10.9
<i>Sparse</i>	SSC	5.4	15.3	7.6
	LRR	4.4	14.9	6.7
	<b>LHR</b>	3.1	13.9	5.6

### C. Stock clustering

It is not trivial to consider applying LHR model to more complicated practical data where the effectiveness of LHR on corrupted data will be over emphasized. In practical world, one of the most difficult data structures to be analyzed is the stock price which can be greatly affected by company news, rumors and global economic atmosphere. Therefore, data mining approaches of financial signals have been proven to be very difficult but on the other hand, it is very profitable.

In this paper, we will discuss how to use the LRR and LHR model to the interesting, albeit not very lucrative, task of stock clustering based on their industrial categories. In many stock exchange centers around the world, stocks are always divided into different industrial categories. For example, on the New York Stock Exchange Center, IBM and J.P.Morgan are respectively categorized into the *computer based system* category and *money center banks* category. It is generally assumed that stocks in the same category always have similar market performance. This basic assumption is widely used by many hedge funds for statistic arbitrage. In this paper, we consider that stocks in the same industrial category span as a subspace and therefore the goal of stock clustering, a.k.a. stock categorization, is to identify a stock's industrial label by its historical prices.

The experiments are conducted on stocks from two global stock exchange markets in New York and Hong Kong. In each market, we choose 10 industrial categories which have the largest market capitalizations. The categories divided by the exchange centers are used as the ground truth label. In each category, we only choose the stocks whose market capitalizations are within the top 10 ranks in one category. The stock prices on New York market are obtained from [45] and the stock prices in Hong Kong market are obtained from [46]. Unfortunately, some historical prices for stocks in [45] are not recorded and provided<sup>3</sup>. Therefore, for the US

<sup>3</sup>For example, in the industrial category of Drug Manufactures, it is not possible to get the historical data of CIPILA.LTD from [45] which is the only the interface for us to get the stock prices in US.

TABLE IV  
CLUSTERING ERRORS OF THE STOCKS IN TEN CATEGORIES FROM NEW YORK AND HONG KONG MARKETS.

Markets	GPCA	RANSAC	LSA	LLMC
New York	60.1	59.3	51.7	54.3
Hong Kong	57.3	55.8	54.7	53.7
Markets	SSC	LRR	LHR	
New York	48.6	44.1	36.2	
Hong Kong	49.1	46.5	38.3	

market, we accumulated 76 stocks divided into 10 classes and each class contains 7 to 9 stocks; for Hong Kong market, we obtain 96 stocks spanning 10 classes. For classification, the weekly closed prices from 07/01/2008 to 31/10/2011 including 200 weeks, are used because financial experts always look at weekly close prices to judge the long-term trend of a price.

As stated previously, the stock prices may have sharp drop and up which cause outliers in the raw data. Besides, the prices of different stocks are various that cannot be evaluated with the same quantity scaler. For the ease of pattern mining, we use the time-based normalization strategy suggested in [47][48] to pre-process the stock prices:

$$\tilde{p}(t) = \frac{p(t) - \mu_\alpha(t)}{\sigma_\alpha(t)},$$

where  $p(t)$  is the price of a certain stock at time  $t$ ,  $\mu_\alpha(t)$  and  $\sigma_\alpha(t)$  are respectively the average value and standard derivation of the stock prices in the interval  $[t - \alpha, t]$ . We plot the normalized stock prices of three categories in Fig.4. After normalization, we further adopt PCA method to reduce the dimensions of stocks from  $\mathbb{R}^{200}$  to  $\mathbb{R}^5$ . Theoretically, the rank of subspaces after PCA should be  $10 - 1 = 9$  because it contains 10 subspaces and the rank is degraded by 1 during PCA implementation. But, in the simulation, we find that the maximal clustering accuracies for both markets are achieved with the PCA dimensions of 5.

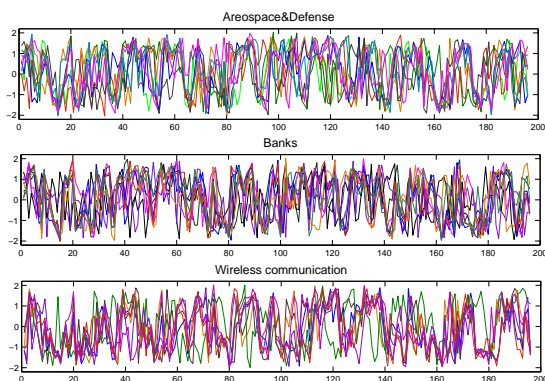


Fig. 4. Normalized stock prices in NY of the categories: Aerospace&Defense, Banks and Wireless communication. In each category, lines in different colors represent different stocks. (best viewed on screen)

The clustering errors of different SC methods on the stocks from these two markets are summarized in Table.IV. From

the results, it is obvious that LHR significantly outperforms other methods. It improves statistic and graph based methods for about 20%. Among all the sparse methods, LHR makes improvements on LRR for about 8%. Although LHR performs the best among all the methods, the clustering accuracy is only about 63% and 61% on US and Hong Kong markets, respectively. The clustering accuracy is not as high as those on the motion data. This may be ascribe to that the raw data and ground truth label themselves contain many uncertainties. See the bottom sub-figure in Fig.4 for the stocks in the *wireless communication* category, the normalized stock marked with the green color performs quite different from other stocks in the same category. But the experimental results reported here is sufficient to verify the effectiveness of subspace clustering for 10 classes categorization. If no intelligent learning approaches were imposed, the expected accuracy may be only 10%. Although with such "bad" raw data, the proposed LHR could achieve the accuracy as high as 62% in a definitely unsupervised way.

## VI. CONCLUSION

This paper presents a log-sum heuristic recovery algorithm to learn the essential low rank structures from corrupted matrices. We introduced a MM algorithm to convert the non-convex objective function a series of convex optimizations via reweighed approaches and proved that the solution may converge to a stationary point. Then, the general model was applied to two practical tasks of LRMR and SC. In both of the two models, we gave the solution/update rules to each variable in the joint optimizations via ADM. For the general PCP problem, LHR extended the feasible region to the boundary of  $\eta + \xi = 0.58$ . For SC problem, LHR achieved state-of-the-art results on motion segmentation and achieved promising results on stock clustering which contain too many outliers and uncertainties. However, a limitation of the proposed LHR model is for the reweighted phenomenon that requires to solve convex optimizations for multiple times. The implementations of LHR is a bit more time consuming than PCP and LRR. Therefore, LHR model is especially recommend to learn the low rank structure from data with denser corruptions and higher ranks.

## APPENDIX

### A. ABBREVIATIONS

**ADM**: Alternating Direction Method, **MM**: Majorization Minimization, **GPCA**:Generalized Principal Component Analysis, **PCP**: Principal Component Pursuit, **LHR**:Log-sum Heuristic Recovery, **LLMC**: Locally Linear Manifold Clustering, **LRMR**: Low Rank Matrix Recovery, **LRR**:Low rank representation, **LRSL**: Low Rank Structure Learning, **LSA**: Local Subspace Affinity, **MoG**: Mixture of Gaussian, **RANSAC**: Random Sample Consensus, **RPCA**:Robust Principal Component analysis, **SC**:Subspace Clustering, **SSC**: Sparse Subspace Clustering.

### B. Convergence of subsequences in the proof of lemma III.4

In this part, we provide the discussions about the properties of the convergent subsequences that are used in the proof of Lemma III.4.

Since sequence  $\hat{X}^t = \{\mathbf{Y}^t, \mathbf{Z}^t, \mathbf{A}^t, \mathbf{E}^t\}$  is generated via Eq.6, we know that  $\hat{X} \in \hat{D}$  strictly holds. Therefore, all the variables (i.e.  $\mathbf{Y}^t, \mathbf{Z}^t, \mathbf{A}^t, \mathbf{E}^t$ ) in set  $\hat{X}$  should be bounded. This claim can be easily verified because that if any variable in the set  $\hat{X}$  goes to infinity, the constraints in domain  $\hat{D}$  will not be satisfied. Accordingly, we know that sequence  $\hat{X}^t$  is bounded. According to the *Bolzano-Welestrass Theorem* [49], we know that every bounded sequence has a convergent subsequence. Since  $\hat{X}^t$  is bounded, it is apparent that there exists a convergent subsequence  $\hat{X}^{t_k}$ . Without the loss of generality, we can construct another subsequence  $\hat{X}^{t_{k+1}}$  which is also convergent. The proof of the convergence of  $\hat{X}^{t_{k+1}}$  relies on the monotonically decreasing property proved in Lemma III.3. Since  $H(\cdot)$  is monotonically decreasing, it is easy to check that  $H(\hat{X}^{t_k}) \geq H(\hat{X}^{t_{k+1}}) \geq H(\hat{X}^{t_{k+2}}) \geq H(\hat{X}^{t_{k+3}}) \geq H(\hat{X}^{t_{k+4}})$ . According to the above inequalities, we get that,

$$\lim_{k \rightarrow \infty} H(\hat{X}^{t_k}) \geq \lim_{k \rightarrow \infty} H(\hat{X}^{t_{k+1}}) \geq \lim_{k \rightarrow \infty} H(\hat{X}^{t_{k+2}}) \quad (30)$$

Since subsequence  $\hat{X}^{t_k}$  converges, it is obvious that  $\lim_{k \rightarrow \infty} H(\hat{X}^{t_k}) = \lim_{k \rightarrow \infty} H(\hat{X}^{t_{k+2}}) = \beta$ . According to the famous *Squeeze Theorem* [50], from (30), we get the  $\lim_{k \rightarrow \infty} H(\hat{X}^{t_{k+1}}) = H(\lim_{k \rightarrow \infty} \hat{X}^{t_{k+1}}) = \beta$ . Since the function  $H(\cdot)$  is monotonically decreasing and  $\hat{X}$  is bounded, the convergence of  $H(\hat{X}^{t_{k+1}})$  can be obtained if and only if the subsequence  $\hat{X}^{t_{k+1}}$  is convergent.

### REFERENCES

- [1] X. Li and Y. Pang, "Deterministic column-based matrix decomposition," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 1, pp. 145–149, 2010.
- [2] Y. Yuan, X. Li, Y. Pang, X. Lu, and D. Tao, "Binary sparse nonnegative matrix factorization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 772–777, 2009.
- [3] S. Hu and J. Wang, "Absolute exponential stability of a class of continuous-time recurrent neural networks," *Neural Networks, IEEE Transactions on*, vol. 14, no. 1, pp. 35–45, jan 2003.
- [4] A. Goldberg, X. J. Zhu, B. Recht, J. Sui, and R. Nowak, "Transduction with Matrix Completion: Three Birds with One Stone," *NIPS*, 2010.
- [5] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
- [6] S. Hu and J. Wang, "Quadratic stabilizability of a new class of linear systems with structural independent time-varying uncertainty," *Automatica*, vol. 37, no. 1, pp. 51–59, 2001.
- [7] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices," *NIPS*, 2009.
- [8] Y. Deng, Y. Liu, Q. Dai, Z. Zhang, and Y. Wang, "Noisy depth maps fusion for multiview stereo via matrix completion," *submitted to IEEE journal of selected topics in signal processing*.
- [9] D. Tao, X. Li, X. Wu, and S. Maybank, "Geometric mean for subspace selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 260–274, feb. 2009.
- [10] Y. Deng, Q. Dai, R. Wang, and Z. Zhang, "Commute time guided transformation for feature extraction," *Computer Vision and Image Understanding*, vol. 116, no. 4, pp. 473–483, 2012.
- [11] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [12] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.
- [13] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 59, no. 3, pp. 1–37, May 2011.
- [14] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *Tech. Rep. arXiv:0906.2220*, Jun 2009.
- [15] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with outliers," *arXiv:1011.1518v3*, 2010.
- [16] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *International Conference on Machine Learning*, 2010, pp. 663–670.
- [17] J. Cai, E. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *Preprint*, 2008.
- [18] E. J. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted  $\ell_1$  minimization," *J. Fourier Anal. Appl.*, pp. 877–905, 2007.
- [19] M. Fazel, "Matrix rank minimization with applications," *Ph.D. thesis, Stanford University*, March 2002.
- [20] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 1–59, 2000.
- [21] A. Ganesh, J. Wright, X. Li, E. J. Candes, and Y. Ma, "Dense error correction for low-rank matrices via principal component pursuit," *Proceedings of International Symposium on Information Theory*, June 2010.
- [22] R. Vidal, "Subspace clustering," *Signal Processing Magazine, IEEE*, vol. 28, no. 2, pp. 52–68, 2011.
- [23] C.-S. Foo, C. B. Do, and A. Y. Ng, "A majorization-minimization algorithm for (multiple) hyperparameter learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, 2009, pp. 321–328.
- [24] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, "Majorization minimization algorithms for wavelet-based image restoration," *Image Processing, IEEE Transactions on*, vol. 16, no. 12, pp. 2980–2991, dec. 2007.
- [25] J. Bioucas-Dias, M. Figueiredo, and J. Oliveira, "Total variation-based image deconvolution: a majorization-minimization approach," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2, may 2006, p. II.
- [26] M. Fazel, "Matrix rank minimization with applications," *PhD thesis, Stanford University*, 2002.
- [27] M. Fazel, H. Hindi, and S. Boyd, "Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, june 2003, pp. 2156–2162 vol.3.
- [28] K. Mohan and M. Fazel, "Reweighted nuclear norm minimization with application to system identification," in *Proceedings of American Control Conference*, 2010.
- [29] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J.R.Statistic.Soc.B*, vol. 67, pp. 301–320, 2005.
- [30] D. Hunter and R. Li, "Variable selection using mm algorithms," *Annals of statistics*, vol. 33, no. 4, p. 1617, 2005.
- [31] K. Lange, "A gradient algorithm locally equivalent to the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 425–437, 1995.
- [32] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Image Science*, vol. 2, no. 1, pp. 183–202.
- [33] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–123, 2010.
- [34] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Tech. Rep. arXiv:1009.5055v2*, Mar 2011.
- [35] M. Lees, "A note on the convergence of alternating direction methods," *Mathematics of Computation*, vol. 16, no. 77, pp. 70–75, 1963.
- [36] D. Donoho, "De-noising by soft-thresholding," *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 613–627, May 1995.
- [37] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," in *Found. of Comput. Math.*, 2008.
- [38] "http://perception.csl.uiuc.edu/matrix-rank/sample\_code.html."
- [39] C. Benedek and T. Sziranyi, "Bayesian foreground and shadow detection in uncertain frame rate surveillance videos," *Image Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 608–621, April 2008.
- [40] J. K. Suhr, H. G. Jung, G. Li, and J. Kim, "Mixture of gaussians-based background subtraction for bayer-pattern image sequences," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 3, pp. 365–370, 2011.
- [41] "http://en.wikipedia.org/wiki/type\_i\_and\_type\_ii\_errors."

- [42] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981. [Online]. Available: <http://doi.acm.org/10.1145/358669.358692>
- [43] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, june 2003, pp. I–621 – I–628 vol.1.
- [44] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.
- [45] "Yahoo!finacial." in <http://finance.yahoo.com>.
- [46] "Google finacial." [Online]. Available: <http://www.google.com.hk/finance?q=>
- [47] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani, "Mining the stock market (extended abstract): which measure is best?" in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 487–496.
- [48] T. Wittman, "Time-series clustering and association analysis of financial data," 2002.
- [49] "Bolzano-welestrass theoreme." [Online]. Available: [http://en.wikipedia.org/wiki/Bolzano%E2%80%93Weierstrass\\_theorem](http://en.wikipedia.org/wiki/Bolzano%E2%80%93Weierstrass_theorem)
- [50] "Sequezz theoreme." [Online]. Available: [http://en.wikipedia.org/wiki/Squeeze\\_theorem](http://en.wikipedia.org/wiki/Squeeze_theorem)